HAL open science

Dynamisme et Domination

Swan Dubois, Mohamed Hamza Kaaouachi, Franck Petit

To cite this version:

Swan Dubois, Mohamed Hamza Kaaouachi, Franck Petit. Dynamisme et Domination. ALGOTEL 2015 - 17èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Jun 2015, Beaune, France. hal-01145496

HAL Id: hal-01145496

https://hal.science/hal-01145496

Submitted on 24 Apr 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dynamisme et Domination †

Swan Dubois[1,2,3], Mohamed-Hamza Kaaouachi[1,2,3] et Franck Petit[1,2,3]

[1] *Sorbonne Universités, UPMC Université Paris 6, F-75005, Paris, France*
[2] *CNRS, UMR 7606, LIP6, F-75005, Paris, France*
[3] *Inria, Équipe-projet REGAL, F-75005, Paris, France*

Dans cet article, nous traitons de la construction d'un ensemble dominant minimal dans un système réparti hautement dynamique. En particulier, les hypothèses de connexité faites sur le système sont très faibles : nous supposons uniquement que tout processeur est capable de communiquer infiniment souvent avec tout autre (pas nécessairement de manière directe, ni par la même route). Dans ce contexte, nos contributions sont une nouvelle définition de l'ensemble dominant minimal applicable dans ce modèle et une condition nécessaire et suffisante sur la topologie du système pour permettre l'existence d'une solution au problème.

Keywords: Ensemble dominant minimal, Système réparti dynamique, Graphe variant dans le temps

1 Introduction

Les communications sans fils se sont énormément développées amenant avec elles de nombreuses et nouvelles possibilités. Les participants sont susceptibles de rejoindre, de quitter ou de se déplacer dans de tels réseaux à des instants imprévisibles. La caractéristique commune de ces réseaux est leur forte dynamique, ce qui signifie que leur topologie ne cesse d'évoluer dans le temps. Il est donc nécessaire de concevoir et d'étudier des modèles théoriques capables de décrire le plus fidèlement cette nouvelle caractéristique. Dans cet article, nous adoptons le modèle des *graphes variant dans le temps* proposé dans [2] (*cf.* section 2 pour plus de détails).

Nous nous intéressons au problème de l'*ensemble dominant minimal*. Rappelons qu'un ensemble dominant est un sous-ensemble des sommets d'un graphe tel que chaque sommet du graphe qui n'est pas dans l'ensemble dominant est voisin d'au moins un sommet de l'ensemble dominant. Un tel ensemble est minimal si aucun de ses sous-ensembles stricts n'est lui-même dominant. Comme de nombreux autres problèmes issus de la théorie des graphes comme la construction d'arbre couvrant, le coloriage, etc., la construction d'un ensemble dominant minimal est utile dans le cadre de l'algorithmique répartie en tant que brique élémentaire pour développer des protocoles plus élaborés (routage hiérarchique, clustering, diffusion,...).

La difficulté de définir les problèmes de couverture (comme la construction de l'ensemble dominant minimal) dans un contexte dynamique a été soulignée dans [1]. En effet, les auteurs ont montré que la définition classique de ces problèmes peut devenir ambiguë, incorrecte, voire inadaptée en raison de la dynamique du système. Par exemple, si la dynamique du graphe est modélisée par une suite de graphes statiques et qu'un nouvel ensemble dominant minimal est recalculé à chaque changement topologique [4], la stabilité de cet ensemble dominant dépend de la fréquence de ces changements topologiques. Cette définition peut donc conduire au calcul d'un ensemble très instable difficile à exploiter.

C'est pourquoi notre première contribution dans cet article est de proposer une nouvelle définition de l'ensemble dominant minimal adaptée aux graphes variant dans le temps munis d'hypothèses très faibles sur leur connexité, par conséquent, potentiellement hautement dynamiques. La seconde contribution est une condition nécessaire et suffisante sur la topologie de ces graphes variant dans le temps assurant l'existence d'une solution déterministe au problème de la construction d'un ensemble dominant minimal (*cf.* section 3 pour plus de détails). En raison de l'espace limité, nous ne présenterons que les points clés de ce travail ‡.

†Ce travail été effectué dans le cadre du Labex SMART et a bénéficié d'une aide de l'Etat gérée par l'Agence Nationale de la Recherche au titre du programme Investissements d'Avenir portant la référence ANR-11-LABX-65.

‡. Pour plus de détails, le lecteur peut se référer au rapport de recherche [3].

2 Modèle

Nous rappelons ici le modèle introduit dans [2] pour décrire les systèmes répartis dynamiques. Nous considérons un système réparti asynchrone par passage de messages constitué d'un ensemble V fixe de n processeurs. Les envois de messages entre les processeurs ont lieu sur un ensemble d'arêtes E dont la disponibilité évolue au cours de la durée de vie $\mathcal{T} \subseteq \mathbb{T}$ du système (avec $\mathbb{T} = \mathbb{N}$ ou $\mathbb{T} = \mathbb{R}^+$).

Définition 1 (Graphe variant dans le temps [2]) *Un graphe variant dans le temps (GVT) g est un tuple $(V, E, \mathcal{T}, \rho, \zeta, \phi)$ où V est un ensemble de processeurs, $E \subseteq V \times V$, $\rho : E \times \mathcal{T} \to \{0,1\}$ (appelée fonction de présence) indique si une arête donnée est présente à un instant donné, $\zeta : E \times \mathcal{T} \to \mathbb{T}$ (appelée fonction de latence des arêtes) indique le temps nécessaire à un message pour traverser une arête donnée s'il est émis à un instant donné et $\phi : V \times \mathcal{T} \to \mathbb{T}$ (appelée fonction de latence des processeurs) indique le temps nécessaire à un calcul local d'un processeur donné à un instant donné.*

Pour un GVT g, $\mathcal{T}_g = \{t_0, t_1, \ldots\}$ est le sous-ensemble de $\mathcal{T}$ qui regroupe l'ensemble des instants pour lesquels g subi un changement topologique. L'évolution de g au cours de sa durée de vie $\mathcal{T}$ peut être décrite par la suite de graphes $\mathcal{S}_g = g_1, g_2, \ldots$, où $g_i = (V, E_i)$ satisfait $e \in E_i$ si et seulement si $\forall t \in [t_i, t_{i+1}[, \rho(e,t) = 1$. L'absence ou la présence des arêtes sont supposées être instantanément détectées par les processeurs adjacents. Chaque processeur dispose d'une primitive de communication non bloquante nommée **Envoyer_Répéter** qui assure la propriété suivante : lorsqu'un processeur p invoque **Envoyer_Répéter**(m,q) (où m est un message et $q \in V$) à l'instant t, le message m sera délivré à q en un temps fini s'il existe un instant $t' \geq t$ tel que l'arête $\{p,q\}$ soit présente à l'instant t' durant au moins $\zeta(\{p,q\},t')$ unités de temps, *i.e.*, l'arête doit être présente suffisamment longtemps pour surmonter le délai de communication.

L'*état* d'un processeur est défini par la valeur de ses variables. Pour un GVT g, une *configuration* de g sera un $n+2$-uple $(g_i, M_i, p_1, p_2, \ldots, p_n)$ tel que $g_i \in \mathcal{S}_g$, M_i est l'ensemble des multi-ensembles de messages en transit entre les processeurs et p_1,...,p_n est l'état des n processeurs de V. Certaines variables sont distinguées dans la spécification du problème comme des variables de *sortie*. Un processeur p *retourne* la valeur v dans une configuration γ si une de ses variables de sortie possède la valeur v dans γ.

Une *exécution* du système réparti modélisé par un GVT g est une suite de configurations $e = \gamma_0, \ldots, \gamma_k, \gamma_{k+1}, \ldots$ telle que pour tout $k \geq 0$, un des événements suivants se produit durant l'étape (γ_k, γ_{k+1}) : un changement topologique ou (ii) au moins un processeur a reçu un message, envoyé un message ou exécuté une action interne pour modifier son état. L'*algorithme* exécuté par g décrit l'ensemble des actions internes autorisées pour les processeurs (en fonction de leur état actuel et d'événements extérieurs comme la réception d'un message ou l'expiration d'un délai). Lorsqu'une arête disparaît, tous les messages en transit sur cette arête sont perdus. Lorsqu'une arête apparaît, elle ne contient pas de messages en transit.

Les auteurs de [2] ont montré que la notion classique de chemin dans un graphe est trop forte dans un GVT. En effet, des processeurs peuvent communiquer même s'il n'existe pas à tout instant un chemin statique entre eux. Pour réaliser une communication entre deux processeurs, l'existence d'un *chemin temporel* est suffisante. Un chemin temporel entre deux processeurs p et q est une suite d'arêtes adjacentes reliant p à q telle que la disponibilité des arêtes, la latence des arêtes et la latence des processeurs rendent possible l'envoi d'un message de p à q en utilisant la primitive **Envoyer_Répéter** de proche en proche.

En se basant sur les propriétés requises sur les chemins temporels (comme la périodicité, la symétrie...), les auteurs de [2] proposent une classification intéressante des GVT. Dans cet article, nous nous intéressons à la famille présentant des contraintes minimales sur la connexité du GVT.

Définition 2 (GVT temporellement connexe [2]) *Un GVT $(V, E, \mathcal{T}, \rho, \zeta, \phi)$ est temporellement connexe si, pour tout instant $t \in \mathcal{T}$ et pour toute paire de processeurs p et q de V, il existe un chemin temporel de p vers q après l'instant t. La famille des GVT temporellement connexes est notée $\mathcal{TC}$.*

Cette définition implique que la durée de vie d'un GVT temporellement connexe est infinie. Notons qu'un GVT temporellement connexe peut être non connexe à tout instant et, que dans un tel GVT, la présence d'une arête à un instant donné ne garantit pas que celle-ci soit à nouveau disponible après cet instant. Nous définissons une *arête ultimement absente* comme une arête qui n'apparaît que durant un temps fini pendant la durée de vie du GVT. La principale difficulté rencontrée dans l'écriture d'algorithmes pour les GVT de la famille $\mathcal{TC}$ est de gérer ces arêtes ultimement absentes étant donné qu'aucun processeur n'est capable de prédire si une arête adjacente est une arête ultimement absente ou pas.

Définition 3 (Graphe sous-jacent (ultime)) *Étant donné un GVT $g = (V, E, \mathcal{T}, \rho, \zeta, \phi)$, le graphe sous-jacent de g est le graphe $U_g = (V, E)$. Le graphe sous-jacent ultime de g est le graphe partiel de U_g défini par $U_g^\omega = (V, E^\omega)$ avec $E^\omega = E \setminus M_g$ où M_g est l'ensemble des arêtes ultimement absentes de g.*

Dans la suite, nous serons amenés à restreindre nos résultats à des sous-familles de GVT particulières définies à partir d'un ensemble de graphes sous-jacents définies comme suit : étant donné un ensemble de graphes $\mathcal{F}$ et une famille de GVT $\mathcal{C}$, la sous-famille de $\mathcal{C}$ induite par $\mathcal{F}$ (notée $\mathcal{C}|_\mathcal{F}$) est l'ensemble de tous les GVT de $\mathcal{C}$ dont le graphe sous-jacent appartient à $\mathcal{F}$.

3 Ensemble Dominant

Spécification. Dans cet article, nous nous intéressons au problème de la construction d'un ensemble dominant minimal dans un GVT de famille $\mathcal{TC}$. Avant toute chose, il est nécessaire de s'attarder sur la spécification du problème dans un cadre dynamique. En effet, deux approches ont déjà été proposées dans [1, 4], mais celles-ci ne sont pas satisfaisantes dans notre contexte, motivant ainsi notre nouvelle définition.

L'approche la plus naturelle est proposée par [4]. Le graphe dynamique est vu comme une suite de graphes statiques et un ensemble dominant minimal est calculé pour chacun de ces graphes (donc à chaque changement topologique). Dans un système très dynamique, l'ensemble dominant calculé risque de ne jamais être stable (si la durée entre deux changements topologiques est plus faible que le temps de calcul de l'ensemble dominant) et donc peu intéressant pour l'application souhaitant l'utiliser.

La seconde approche [1] consiste à calculer un ensemble dominant sur le graphe sous-jacent du GVT. L'avantage de cette approche est la stabilité de l'ensemble calculé. Cependant, sur un GVT de famille $\mathcal{TC}$, l'existence d'arêtes ultimement absentes implique qu'un processeur dans l'ensemble dominé peut ultimement n'avoir aucun voisin dominant, ce qui est contradictoire avec l'intuition d'un ensemble dominant.

Pour remédier à ce problème, nous proposons une définition dans laquelle l'ensemble dominant calculé est stable et dans laquelle tout processeur dans l'ensemble dominé possède au moins infiniment souvent un voisin dominant. En d'autres termes, nous imposons le calcul d'un ensemble dominant minimal sur le graphe sous-jacent *ultime* du GVT. Notons que cette définition généralise la définition classique sur un graphe statique (qui est alors son propre graphe sous-jacent ultime) mais également celle de [1] (si elle est appliquée dans un GVT sans arêtes ultimement absentes). Plus formellement :

Définition 4 (Ensemble dominant minimal temporel) *Un ensemble de processeurs M est un ensemble dominant minimal temporel d'un GVT g si M est un ensemble dominant minimal de U_g^ω.*

Spécification 1 (Ensemble dominant minimal) *Un algorithme $\mathcal{A}$ satisfait la spécification de l'ensemble dominant minimal pour une famille de GVT $\mathcal{C}$ si l'exécution $e = \gamma_0, \gamma_1, \ldots$ de $\mathcal{A}$ sur tout GVT g de $\mathcal{C}$ possède un suffixe $e_i = \gamma_i, \gamma_{i+1}, \ldots$ pour un $i \in \mathbb{N}$ donné tel que tout processeur retourne une unique valeur booléenne dans toute configuration de e_i et que l'ensemble des processeurs retournant* Vrai *est un ensemble dominant minimal temporel de g.*

Impossibilité. Dans un premier temps, nous montrons que ce problème n'admet pas toujours de solution déterministe. Pour cela, nous introduisons une nouvelle définition.

Définition 5 (Ensemble fortement dominant minimal) *Un ensemble fortement dominant minimal d'un graphe g est un sous-ensemble de processeurs de g qui est un ensemble dominant minimal de tout graphe partiel connexe de g.*

Notons que si le graphe sous-jacent d'un GVT $g \in \mathcal{TC}$ admet un ensemble fortement dominant minimal M, alors M est un ensemble dominant minimal temporel de g (car U_g^ω est un graphe partiel connexe de U_g). Intuitivement, notre résultat d'impossibilité est qu'il n'existe pas d'algorithme déterministe satisfaisant la spécification de l'ensemble dominant minimal sur un GVT de famille $\mathcal{TC}$ si le graphe sous-jacent de ce GVT n'admet pas d'ensemble fortement dominant minimal. En effet, comme aucun processeur n'est capable de détecter les arêtes ultimement absentes, l'ensemble dominant minimal calculé par n'importe quel algorithme doit être un ensemble dominant minimal de n'importe quel graphe sous-jacent ultime *potentiel*, donc de tout graphe partiel connexe du graphe sous-jacent. En d'autres termes, l'ensemble dominant minimal calculé est un ensemble fortement dominant minimal du graphe sous-jacent. L'existence d'un tel ensemble est donc une condition nécessaire à l'existence de l'algorithme en question. D'où :

Théorème 1 *Pour tout ensemble de graphes $\mathcal{F}$ contenant au moins un graphe qui n'admet pas d'ensemble fortement dominant minimal, il n'existe pas d'algorithme déterministe qui satisfait la spécification de l'ensemble dominant minimal sur $\mathcal{TC}|_{\mathcal{F}}$.*

Algorithme Nous sommes à présent en mesure de montrer la suffisance de cette condition. Pour cela, nous allons décrire un algorithme satisfaisant la spécification de l'ensemble dominant minimal sur tout GVT de la famille $\mathcal{TC}$ dont le graphe sous-jacent admet un ensemble fortement dominant minimal. Nous commençons par caractériser ces graphes avec le résultat suivant qui découle de la définition 5.

Théorème 2 *Pour tout graphe g et tout ensemble dominant minimal M de g, M est un ensemble fortement dominant minimal de g si et seulement si l'ensemble d'arêtes $\{\{p,q\}|q \in M \cap \mathcal{N}_p\}$ est une coupe de g pour tout processeur $p \in V \setminus M$ où $\mathcal{N}_p$ est l'ensemble des voisins de p dans g.*

Notre algorithme fonctionne de la manière suivante. Chaque processeur construit une copie locale du graphe sous-jacent du GVT. Pour cela, il suffit que chaque processeur envoie sa propre copie locale (initialement vide) à la première apparition d'une arête adjacente, fasse l'union de toutes les copies reçues et propage à nouveau sa copie locale à chaque modification de cette dernière. Une fois la copie locale du graphe sous-jacent calculée, il est possible pour chaque processeur de déterminer s'il appartient à l'ensemble dominant temporel calculé par l'algorithme : le processeur énumère (localement et selon un ordre déterministe commun à tous les processeurs) tous les ensembles dominants minimaux de ce graphe sous-jacent et choisi le premier qui satisfait la propriété du théorème 2 (notez que cela implique une complexité exponentielle en terme de calcul locaux). Il retourne alors *Vrai* ou *Faux* selon qu'il appartienne ou non à cet ensemble dominant. Notons que celui-ci est un ensemble fortement dominant minimal du graphe sous-jacent et donc un ensemble dominant minimal temporel du GVT. Afin d'éviter l'utilisation d'un algorithme de détection de terminaison pour le calcul du graphe sous-jacent, chaque processeur répète ce calcul à chaque actualisation de sa copie locale du graphe sous-jacent. L'existence de cet algorithme permet de déduire :

Théorème 3 *Pour tout ensemble de graphes $\mathcal{F}$ contenant uniquement des graphes qui admettent un ensemble fortement dominant minimal, il existe un algorithme déterministe qui satisfait la spécification de l'ensemble dominant minimal sur $\mathcal{TC}|_{\mathcal{F}}$.*

4 Conclusion

Dans cet article, nous nous sommes concentrés sur le problème de construction d'un ensemble dominant minimal dans un système réparti hautement dynamique. Nous avons tout d'abord proposé une nouvelle définition de ce problème qui semble plus adéquate dans ce contexte que les définitions existantes étant donné qu'elle permet d'obtenir un ensemble stable (donc exploitable par un autre algorithme) et cohérent avec l'intuition d'un ensemble dominant. Nous avons ensuite fourni une condition topologique sur le GVT nécessaire et suffisante pour l'existence d'un algorithme déterministe satisfaisant ce problème. Les perspectives de ce travail sont d'explorer de telles conditions nécessaires et suffisantes dans les autres familles de GVT mais aussi de mener une étude similaire pour les autres problèmes de couverture.

Références

[1] A. Casteigts and P. Flocchini. Deterministic algorithms in dynamic networks : Problems, analysis, and algorithmic tools. Technical report, DRDC 2013-020, 2013.

[2] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *IJPEDS*, 27(5) :387–408, 2012.

[3] S. Dubois, M.-H. Kaaouachi, and F. Petit. Enabling minimal dominating set in highly dynamic distributed systems. Technical report, Inria hal-01111610, 2015.

[4] J. Whitbeck, M. Dias de Amorim, V. Conan, and J.-L. Guillaume. Temporal reachability graphs. In *MobiCom*, pages 377–388, 2012.